

EXHIBIT D

DOCKET NO.: 337722-000122

Filed on behalf of Apple Inc.

By: Larissa S. Bifano, Reg. No. 59,051
James M. Heintz, Reg. No. 41,828
Michael Van Handel, Reg. No. 68,292

DLA Piper LLP (US)
33 Arch Street, 26th Floor
Boston, Massachusetts 02110-1447
Email: Larissa.Bifano@dlapiper.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.
Petitioner

v.

UNILOC 2017 LLC
Patent Owner

IPR2019-00056

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 6,467,088
CHALLENGING CLAIMS 1-21
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
A.	Real Party-in-Interest	1
B.	Related Matters	2
C.	Counsel	2
D.	Service Information, Email, Hand Delivery, and Postal	2
III.	CERTIFICATION OF GROUNDS FOR STANDING	2
IV.	OVERVIEW OF CHALLENGE AND RELIEF REQUESTED	3
A.	Prior Art Patents and Printed Publications	3
B.	Grounds for Challenge	4
V.	OVERVIEW OF THE '088 PATENT	4
A.	Summary of the Alleged Invention	4
B.	Level of Ordinary Skill in the Art	7
C.	Prosecution History	8
VI.	CLAIM CONSTRUCTION	9
A.	“list”	9
VII.	SPECIFIC GROUNDS FOR PETITION	10
A.	Ground I: Claims 1-21 are obvious in view of <i>Cole</i> , <i>MacInnis</i> , and <i>Elgressy</i>	10
1.	Overview of <i>Cole</i>	10
2.	Claims 1, 11, and 21 are obvious in view of <i>Cole</i> , <i>MacInnis</i> , and <i>Elgressy</i>	13
3.	Claims 2 and 12 are obvious in view of <i>Cole</i> , <i>MacInnis</i> , and <i>Elgressy</i>	31
4.	Claims 3 and 13 are obvious in view of <i>Cole</i> , <i>MacInnis</i> , and <i>Elgressy</i>	32

5.	Claims 4 and 14 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	33
6.	Claims 5 and 15 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	34
7.	Claims 6 and 16 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	36
8.	Claims 7 and 17 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	38
9.	Claims 8 and 18 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	39
10.	Claims 9 and 19 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	39
11.	Claims 10 and 20 are obvious in view of <i>Cole, MacInnis,</i> and <i>Elgressy</i>	41
B.	Ground II: Claims 1-21 are unpatentable as being obvious in view of <i>Pitzel, Cole, and Elgressy</i>	42
1.	Overview of <i>Pitzel</i>	42
2.	Claims 1, 11, and 21 are obvious in view of <i>Pitzel, Cole,</i> and <i>Elgressy</i>	44
3.	Claims 2 and 12 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	59
4.	Claims 3 and 13 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	60
5.	Claims 4 and 14 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	61
6.	Claims 5 and 15 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	63
7.	Claims 6 and 16 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	65
8.	Claims 7 and 17 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	66
9.	Claims 8 and 18 are obvious in view of <i>Pitzel, Cole, and</i> <i>Elgressy</i>	66

IPR2019-00056
Patent 6,467,088

10.	Claims 9 and 19 are obvious in view of <i>Pitzel, Cole</i> , and <i>Elgressy</i>	67
11.	Claims 10 and 20 are obvious in view of <i>Pitzel, Cole</i> , and <i>Elgressy</i>	67
VIII.	CONCLUSION	69

I. INTRODUCTION

Apple Inc. (“Apple” or “Petitioner”) respectfully submits this Petition for *Inter Partes* Review of claims 1-21 of U.S. Patent 6,467,088 (“’088 patent” or “’088”) (EX1001). The ’088 patent describes a technique for controlling the reconfiguration of a device in response to a reconfiguration request. *See, e.g.*, ’088 patent, Abstract (EX1001). The technique includes comparing a component required to implement the reconfiguration request and information specifying an additional component currently implemented in the device with a list of known acceptable or unacceptable configurations for the device, and then generating information indicative of an approval or denial of the reconfiguration request based on the result of the comparison. *See, e.g.*, ’088 patent, claim 1 (EX1001). These concepts were well-known long before the ’088 patent was filed, and there was nothing inventive about the concepts at that time.

II. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Apple certifies that Apple is the real party-in-interest, and further certifies that no other party exercised control or could exercise control over the filing of this petition or Apple’s participation in any proceeding instituted on this petition.

B. Related Matters

According to assignment records at the United States Patent and Trademark Office, the '088 patent (EX1001) is currently owned by Uniloc 2017 LLC ("Uniloc"). The '088 patent was asserted against Apple by Uniloc Luxembourg S.A. and Uniloc USA, Inc. in *Uniloc USA, Inc. et al. v. Apple Inc.*, 1:18-cv-00296 (WDTX). That case was filed on April 9, 2018, and was voluntarily dismissed without prejudice on July 19, 2018.

C. Counsel

Lead Counsel: Larissa S. Bifano (Reg. No. 59,051)

Backup Counsel: James M. Heintz (Reg. No. 41,828)

Backup Counsel: Michael Van Handel (Reg. No. 68,292)

D. Service Information, Email, Hand Delivery, and Postal

Apple consents to electronic service at Apple-Uniloc-IPR@dlapiper.com. Petitioner can be reached at DLA Piper LLP (US), 33 Arch Street, 26th Floor, Boston, Massachusetts 02110 and 617-406-6000, Fax: 617-406-6100.

III. CERTIFICATION OF GROUNDS FOR STANDING

Petitioner certifies pursuant to Rule 42.104(a) that the patent for which review is sought is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

IV. OVERVIEW OF CHALLENGE AND RELIEF REQUESTED

Pursuant to Rules 42.22(a)(1) and 42.104(1)-(2), Petitioner challenges claims 1-21 of the '088 patent.

A. Prior Art Patents and Printed Publications

The following references are pertinent to the grounds of unpatentability explained below:¹

1. U.S. Patent No. 5,752,042 (filed June 7, 1996; published May 12, 1998) ("*Cole*" (EX1002)), which is prior art under 35 U.S.C. § 102(b).
2. PCT Application Publication No. WO 97/30549 (published August 21, 1997) ("*MacInnis*" (EX1003)), which is prior art under 35 U.S.C. § 102(b).
3. U.S. Patent No. 6,449,723 (filed October 30, 1998; published September 10, 2002) ("*Elgressy*" (EX1004)), which is prior art under 35 U.S.C. § 102(e).
4. U.S. Patent No. 7,062,765 (filed May 25, 1999; published June 13, 2006) ("*Pitzel*" (EX1005)), which is prior art under 35 U.S.C. § 102(e).

¹ The '088 patent issued from an application filed prior to enactment of the America Invents Act ("AIA"). Accordingly, pre-AIA statutory framework applies.

B. Grounds for Challenge

This Petition, supported by the declaration of Dr. Charles Knutson (“Knutson Declaration” or “Knutson”) (EX1006), requests cancellation of claims 1-21 as unpatentable under 35 U.S.C. § 103. *See* 35 U.S.C. § 314(a).

The grounds for challenge based on the foregoing prior art references include the following:

	Grounds	References	Challenged Claims
1.	§103	<i>Cole</i> in view of <i>MacInnis</i> and <i>Elgressy</i>	1-21
2.	§103	<i>Pitzel</i> in view of <i>Cole</i> and <i>Elgressy</i>	1-21

V. OVERVIEW OF THE '088 PATENT**A. Summary of the Alleged Invention**

The '088 patent relates to controlling the reconfiguration of an electronic device. '088 patent, Abstract (EX1001). The patent acknowledges that, prior to its filing date, a number of techniques existed for updating device components. *Id.*, 1:31-2:14 (EX1001). But according to the patent, these existing techniques failed to account for interdependencies between computer software and/or hardware resources.² The patent asserts that, as a result of these deficiencies, “a need exists

² *Id.*, 1:41-45 (“technique ... cannot efficiently handle ... hardware and software interdependencies”), 1:52-59, 1:65-2:3, 2:10-14 (EX1001).

for improved techniques for managing reconfiguration of electronic devices, such that compatibility determinations can be facilitated.” *Id.*, 2:15-19 (EX1001). Knutson ¶ 31 (EX1006).

Figure 3 of the '088 patent, reproduced below, illustrates an example system for reconfiguring an electronic device. '088 patent, 3:8-10 (EX1001). The system includes a reconfiguration manager (outlined in red), a network (outlined in blue), and an electronic device (outlined in green). *Id.*, 3:8-10, 5:41-54 (EX1001).

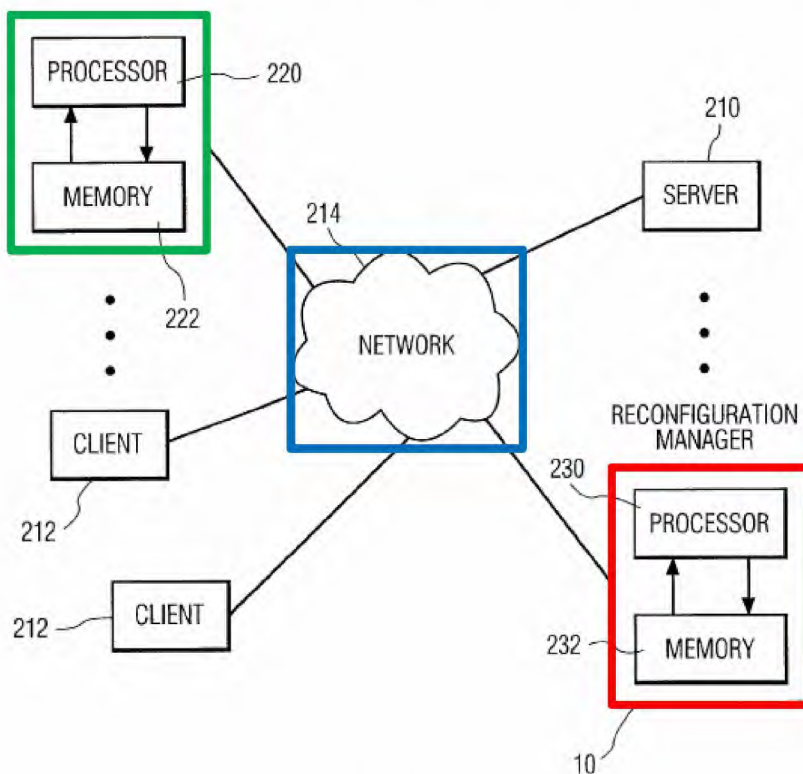


FIG. 3

'088 patent, FIG. 3 (EX1001) (Annotated).

IPR2019-00056
Patent 6,467,088

The electronic device includes a number of software and/or hardware components. '088 patent, 2:59-64, 3:20-24, 6:30-35 (EX1001). The reconfiguration manager includes a list of known configurations, which “indicat[e] which of a set of software components ... are known to work well together or are otherwise compatible.” *Id.*, 3:39-42 (EX1001). This “list” can be “any stored representation of information indicative of component compatibility.” *Id.*, 4:6-8 (EX1001). For example, the list can be implemented in a graphical form, “as a stored table, set of tables, or other type of list in a memory of the reconfiguration manager ..., as a po[r]tion of a program executed by the reconfiguration manager ..., or in any other suitable format.” *Id.*, 3:64-4:2 (EX1001). Knutson ¶ 32 (EX1006).

The '088 patent provides an example of an electronic device that stores “version 1.1 of a software component A, version 2.3 of a software component B, and version 2.0 of a software component C.” '088 patent, 3:20-25 (EX1001). Using this example, the patent describes how the reconfiguration manager can receive a request indicating that the device “wants to upgrade ... to include version 2.0 of software component A.” *Id.*, 4:12-15 (EX1001). The request can include a list of components already on the device. *Id.*, 4:15-19 (EX1001). Knutson ¶ 33 (EX1006).

The reconfiguration manager then processes the request, and determines whether the requested upgrade is compatible with other components of the device. '088 patent, 4:27-31 (EX1001). Using the above example, the reconfiguration manager determines whether the “requested upgrade, in this case version 2.0 of component A, is compatible with other components of [the] device ... i.e., version 2.3 of component B and version 2.0 of component C.” *Id.*, 4:28-32 (EX1001). Depending on the result of this determination, the upgrade request is either approved as compatible with the current configuration of the device, or denied as incompatible with the current configuration of the device. *Id.*, 5:4-29 (EX1001). If compatible, the requested upgrade is delivered from the reconfiguration manager to the device as part of the response. *Id.*, 4:33-37 (EX1001). Knutson ¶ 34 (EX1006).

As the cited prior art demonstrates, at the time of the alleged invention, the problems associated with upgrade compatibility were well-known, and multiple solutions to these problems had been developed. Knutson ¶ 35 (EX1006). As the discussion below demonstrates, the purported invention of the '088 patent provided nothing new, and only rehashed known techniques for upgrading devices. *Id.*

B. Level of Ordinary Skill in the Art

A POSA for the '088 patent would have had at least a Bachelor's Degree in computer science, computer engineering, or a related subject or the equivalent, and

two years of experience working with databases, computer networks, and related technologies. Knutson ¶ 37 (EX1006).

C. Prosecution History

The '088 patent was filed as U.S. Patent Application No. 09/343,607 ("'607 application") on June 30, 1999. During prosecution, the Patent Office initially issued an Office Action that rejected all 21 claims under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,301,707 to Carroll et al. ("*Carroll*"). File History ("FH") of '607 application, Office Action of 02/14/2002, 2-5 (EX1007).

In response, the applicant argued that *Carroll* failed to teach "(1) receiving information representative of a reconfiguration request relating to the electronic device and (2) comparing the determined component and information specifying at least one additional component currently implemented in the electronic device with at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device." FH of '607 application, Response of May 7, 2002, 3 (EX1008).

The Patent Office then issued a Notice of Allowance. FH of '607 application, Notice of Allowance of 07/29/2002 (EX1009). The Examiner agreed with the applicant that *Carroll* "does not teach or suggest comparing an additional component with one of the list[s] in response to a request." *See id.* The Examiner

stated that, for this reason, “the claims are allowable over the art of record.” *See id.* As further described herein, the element the Examiner found to be allowable (“comparing an additional component with one of the list[s] in response to a request”) was well-known at the time of the alleged invention.

VI. CLAIM CONSTRUCTION

At the time of the filing of this Petition, claim terms of an unexpired patent in *inter partes* review are given the “broadest reasonable construction in light of the specification.” 37 C.F.R. § 42.100(b); *In re Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2142 (2016). However, if *inter partes* review of the ’088 patent is instituted, the patent is likely to expire before a Final Written Decision is issued in the proceeding. Claim terms of an expired patent in *inter partes* review are construed in accordance with the claim construction standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005). The construction proposed for the term below should be applied regardless of whether the term is interpreted under the *Phillips* standard or the “broadest reasonable interpretation” standard.

A. “list”

The term “list” is defined by the ’088 patent as, and should be interpreted to include, “any stored representation of information indicative of component compatibility.” Knutson ¶ 46 (EX1006). The ’088 patent explicitly states that “[t]he term ‘list’ as used herein is [] intended to include any stored representation

IPR2019-00056
Patent 6,467,088

of information indicative of component compatibility.” ’088 patent, 4:6-8 (EX1001). The patent further states that a list can be implemented “in graphical form,” “as a stored table,” as a “set of tables,” as an “other type of list,” as a “po[r]tion of a program,” or in “any other suitable format.” *Id.*, 3:64-4:2 (EX1001). A list can indicate “pair-wise compatibility among components” or can “include information indicative of compatibility between groups of multiple components.” *Id.*, 4:2-6 (EX1001). Knutson ¶ 46 (EX1006).

VII. SPECIFIC GROUNDS FOR PETITION

Pursuant to Rule 42.104(b)(4)-(5), the following sections (as confirmed in Knutson ¶¶ 47-218 (EX1006)) detail the grounds of unpatentability, the limitations of the challenged claims of the ’088 patent, and how these claims were therefore obvious in view of the prior art.

A. Ground I: Claims 1-21 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

Cole, *MacInnis*, and *Elgressy* were not considered during prosecution, but are highly relevant to the claims of the ’088 patent.

1. Overview of *Cole*

Cole discloses a system that “selects code updates to download to a client computer.” *Cole*, Abstract (EX1002). As shown in the figure below, the system includes a selection server (outlined in red), a client (outlined in green), and a

network (i.e., Internet) between the selection server and the client. *Id.*, 2:15-19, 2:32-38 (EX1002).

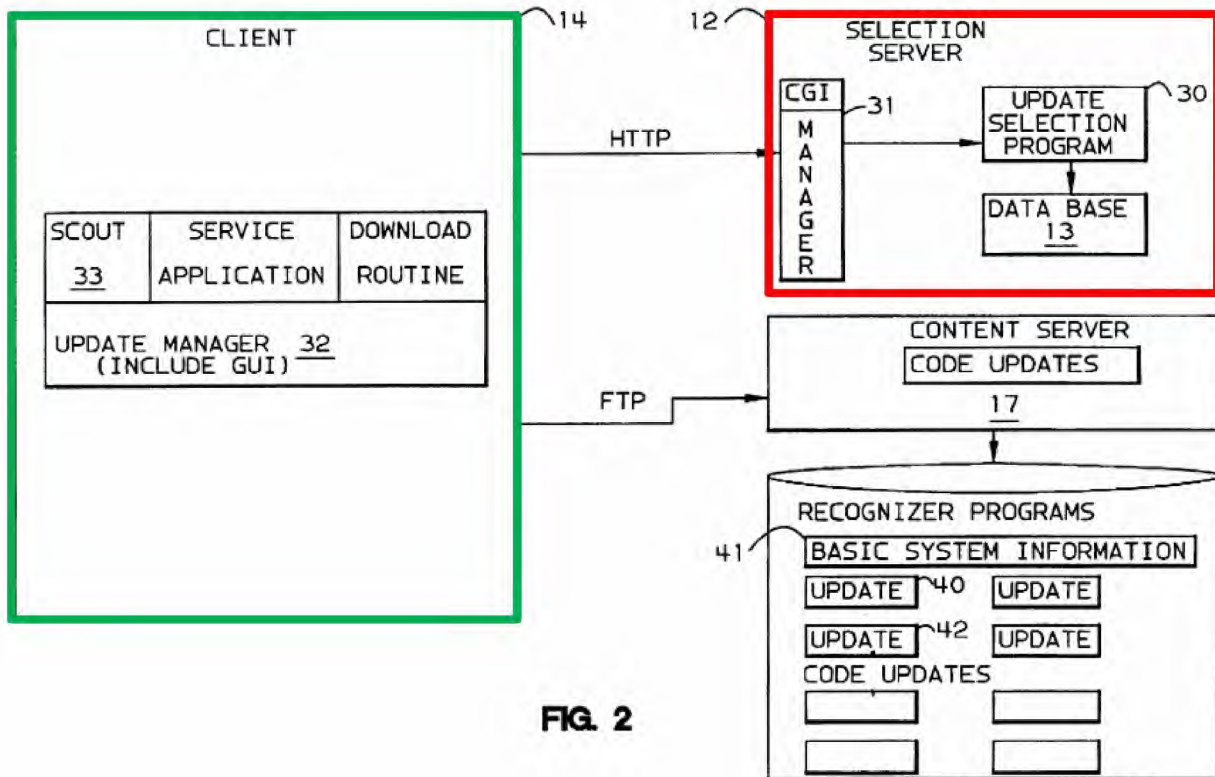


FIG. 2

***Cole*, FIG. 2 (EX1002) (annotated).**

The selection server “automat[es] selection of code updates” for downloading to the client. *Cole*, Abstract, 2:52-56 (EX1002). A database in the selection server stores metadata identifying system characteristics required for each of the code updates. *Cole*, 2:57-59, 3:56-4:47, FIG. 2 (EX1002). This metadata is used to identify which code updates are potentially appropriate for which configurations of a client. *Id.*, 3:56-4:47, FIGs. 3(a), 3(b) (EX1002). Knutson ¶ 49 (EX1006).

After a user “selects an icon” on the client to invoke an update manager (*Cole*, 3:9-16 (EX1002)), the client sends basic system information about the client, such as the “system model, pre-load software level, BIOS level, and ... type of operating system” to the selection server (*Cole*, 3:45-55 (EX1002)). The selection server then determines which code updates are potentially appropriate for the client and which code updates are inappropriate for the client by correlating the metadata in the selection server’s database to the basic system information obtained from the client, where matches “indicate that the corresponding code updates are potentially appropriate for [the] client.” *Id.*, 3:56-4:45 (EX1002). If a code update’s system requirements do not match the client’s basic system information, the code update is eliminated from further consideration. *Id.*, 4:30-34, 4:41-45 (EX1002). If a code update’s system requirements do match the client’s basic system information, the selection server sends the client “FTP addressing information of a corresponding recognizer program” and the client downloads the recognizer program. *Id.*, 4:25-30, 4:35-41, 4:45-47 (EX1002). This recognizer program then “further investigates the hardware, software and other components of the client” to further “determin[e] whether the corresponding code update is appropriate for the client.” *Id.*, 5:2-6 (EX1002). If it is, the client can “download[]

the code update[] from the content server.” *Id.*, 6:27-30 (EX1002). Knutson ¶ 50 (EX1006).

2. Claims 1, 11, and 21 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) Claims 1, 11 - “[a] processor-implemented method)/[a]n apparatus]]³ for controlling the reconfiguration of an electronic device”

Claim 21 - “[a]n article of manufacture comprising a machine-readable medium containing one or more software programs”

Cole discloses a selection server that “selects code updates to download to a client computer.” *Cole*, Abstract (EX1002). The selection server can include a “computer workstation, associated programming and a modem.” *Id.*, 2:38-40 (EX1002). The selection server also includes a “‘CGI’ program,” an “update selection program,” and a “database.” *Id.*, 3:7-9, 3:56-4:25, FIG. 2 (EX1002). The selection server is therefore an apparatus, and includes a processor that executes a

³ Several claims of the ’088 patent contain nearly identical limitations. Petitioner addresses these limitations together. Where headings correspond to multiple claims, Petitioner represents any differences between the language of the claims in brackets. Where substantive differences exist in the language of the claims, those substantive differences are addressed in detail in the section corresponding to the heading.

software program stored in memory to implement a method. Knutson ¶ 55 (EX1006).

The selection server identifies code updates which are appropriate for the client given the client's basic system information, and provides the client with addresses of these code updates so they can be downloaded. *See Cole*, 1:45-65, 3:47-4:45 (EX1002). The selection server also removes code updates that are inappropriate for the client from further consideration. *See id.*, 3:56-62, 4:30-34 (EX1002). Knutson ¶ 56 (EX1006).

In facilitating the installation of appropriate code updates on the client ("electronic device"), while preventing the installation of inappropriate code updates, the selection server "control[s] the reconfiguration of an electronic device." Knutson ¶¶ 54, 57 (EX1006).

b) "receiv[[ing]/[e]] information representative of a reconfiguration request relating to [[the]/[an]] electronic device"

After a user selects an icon on the client to invoke an update manager, "the update manager [on the client] contacts [] general manager 31 [on the selection server] to begin a session" for selecting and downloading updates. *Cole*, 3:12-18 (EX1002). Because this contact initiates a session to update the client, the contact is "information representative of a reconfiguration request relating to the electronic device." Knutson ¶ 59 (EX1006).

In addition to contacting the selection server, the update manager sends information about versions of applications already on the client, such as the update manager, a scout, a service application, and a download routine. *Cole*, 3:14-20 (EX1002). Because this information is sent in response to the user having invoked the update manager, this information is also “information representative of a reconfiguration request relating to the electronic device.” Knutson ¶ 60 (EX1006).

A recognizer program is then downloaded and run on the client to obtain basic system information, such as the client’s “system model, pre-load software level, BIOS level, and ... type of operating system.” *Cole*, 3:45-50 (EX1002). The client sends the basic system information to the selection server, which “initiates the selection update program” at the selection server to determine which code updates are potentially appropriate for the client and which code updates are inappropriate for the client. *Id.*, 3:52-4:45 (EX1002). Since the basic system information is used to identify code updates that are potentially appropriate for the client, the basic system information is also “information representative of a reconfiguration request relating to the electronic device.” Knutson ¶ 61 (EX1006).

In sum, *Cole* discloses that the client (i) contacts the selection server, (ii) sends information about versions of applications already on the client to the selection server, and (iii) sends basic system information to the selection server,

each of which is “information representative of a reconfiguration request relating to the electronic device.” Knutson ¶¶ 58, 62 (EX1006).

c) “determin[[e]/[ing]] at least one device component required to implement the reconfiguration request”

Cole discloses that a user invokes the update manager to prompt the selection server to identify and download code updates. *See Cole*, 3:12-16, 6:23-26, 6:46-55 (EX1002). A code update is a “device component required to implement the reconfiguration request.” *See Cole*, Abstract, 3:40-4:47 (EX1002). Knutson ¶ 64 (EX1006).

Cole also discloses that the database in the selection server stores metadata describing basic system requirements for each of the code updates. *Cole*, 3:1-6, 3:56-4:25 (EX1002). The system requirements can include, for example, a required Bios level, Bios date, pre-load level, or motherboard ID. *Id.*, 2:52-59, 3:56-4:47 (EX1002). Any one of these system requirements listed in the metadata is also a “device component required to implement the reconfiguration request,” because if the basic system information of the client does not meet one of these requirements, the corresponding code update is eliminated from further consideration and is not installed on the client. *See id.*, 3:56-4:1, 4:30-34, 4:41-45 (EX1002). Knutson ¶ 65 (EX1006).

The selection server “determines” the code updates and “determines” the system requirements by accessing a database table, which lists code updates and metadata for the code updates. *Cole*, 4:1-24 (EX1002). Knutson ¶ 66 (EX1006). The code updates and basic system requirements are also “determined” by the selection server when information about the code updates and the metadata for the code updates is written into the database. *See Cole*, 3:1-6, 3:56-4:25 (EX1002). Knutson ¶ 66 (EX1006).

Thus, *Cole* discloses that the selection server determines a (i) code update and (ii) system requirement for a code update, each of which is a “device component required to implement the reconfiguration request.” Knutson ¶¶ 63, 67 (EX1006).

To the extent a narrow interpretation of “device component required to implement the reconfiguration request” is taken, requiring that the device component be a particular component requested as part of the reconfiguration request,⁴ these features were well-known at the time of the alleged invention. Knutson ¶ 68 (EX1006).

⁴ During prosecution, the Examiner appears to have interpreted the claimed “determined [device] component” to be a “requested” component. *See FH of ’607*

For example, *MacInnis* discloses a “system[] for downloading software and data modules into terminals over a network” (*MacInnis*, 1:5-7 (EX1003)), where a user selects a particular software module to download (e.g., the video game “Mortal Combat”), and a table is checked to determine whether the selected software module is compatible with the user’s terminal (*MacInnis*, 13:21-14:10, 14:16-15:9, 15:15-16:10, 16:21-17:18, FIG. 3A (EX1003)). Knutson ¶ 69 (EX1006).

Like *Cole*, *MacInnis* discloses use of a table that identifies compatibility requirements for particular software modules. *See MacInnis*, 7:10-13, 8:3-5, 9:17-22, 10:7-11, 10:19-12:11, FIG. 3A (EX1003). And like *Cole*, *MacInnis* discloses that characteristics of a terminal can be compared with the compatibility requirements of a software module using a table (e.g., the table of FIG. 3A) to determine whether the terminal can download a particular software module. *See id.*, 8:15-20, 9:4-14, 12:19-13:5, 15:15-16:13, FIGs. 2, 3A, 3B (EX1003). Knutson ¶ 70 (EX1006).

MacInnis discloses that a user of a terminal can request a particular software module for installation on the terminal, such as the video game “Mortal Combat.”

application, Notice of Allowance of 07/29/2002 (EX1009). Petitioner adds *MacInnis* to address the Examiner’s apparent interpretation of the claim language.

MacInnis, 13:21-22, 16:21-17:5 (EX1003). In response to the user selection, versions of that particular software module (i.e. “Mortal Combat”) and corresponding compatibility requirements are identified in the table (e.g., the table of FIG. 3A). *Id.*, 13:22-14:10, 17:5-10, FIGs. 3A, 5 (EX1003). The compatibility requirements associated with the versions of the requested software module in the table are then compared with the characteristics of the terminal to determine whether any of the versions are compatible with the terminal. *Id.*, 14:3-7, FIGs. 3A, 3B (EX1003). Knutson ¶ 71 (EX1006).

To a POSA at the time of the alleged invention, it would have been obvious to modify *Cole* so that the selection server receives a request for a particular code update, and identifies that particular code update and its corresponding system requirements in the database table for comparison with the basic system information of the client, in light of the teachings of *MacInnis*. Knutson ¶ 72 (EX1006). A POSA would have been motivated to make this modification to decrease the number of processing operations required to identify whether a particular piece of software is compatible with a client. *Id.* For example, such a modified system would have identified whether a particular code update is compatible with the client, rather than having to determine whether each code update (even those for which the user may not be interested) in the database table

would be compatible with the client. *Compare MacInnis*, 13:21-14:3 (“download mechanism ... [then] locates the two versions of the requested ‘Mortal Combat’ video game in the table”) (EX1003) *with Cole*, 3:57-62 (determining “which code updates are consistent with the basic system information ... and which code updates are inconsistent”) (EX1002). If a user were interested in a particular code update, identifying whether that particular code update is compatible with the client would be more efficient and would require less processing than determining whether each code update in the database table is compatible with the client. Knutson ¶¶ 63, 72 (EX1006).

d) “compar[[ing]/[e]] the determined component and information specifying at least one additional component currently implemented in the electronic device with at least one of [[a]/[the]] list of known acceptable configurations for the electronic device and [[a]/[the]] list of known unacceptable configurations for the electronic device”

As discussed above, each of a (i) code update and (ii) system requirement (e.g., pre-load level, BIOS level, motherboard ID), as disclosed by *Cole*, is a determined “device component required to implement the reconfiguration request.” *See* Section VII.A.2.c, *supra*. Knutson ¶ 74 (EX1006).

Cole discloses that a client sends basic system information to the selection server. *Cole*, 3:45-55 (EX1002). Any one of the client’s system model, pre-load software level, BIOS level, type of operating system, or motherboard ID, as described in the basic system information, is an “additional component currently

IPR2019-00056
Patent 6,467,088

implemented in the electronic device.” Knutson ¶ 75 (EX1006). Thus, the basic system information sent from the client to the selection server is “information specifying at least one additional component currently implemented in the electronic device,” as claimed. *Id.*

Cole provides an example table, shown below, that lists code updates and the basic system requirements metadata for these code updates.

Code Updates	Basis System Information/ Meta Data
Device Driver ABCDE.DRV	BIOS Level 123 BIOS date 1/1/96 Mother Board ID XYZ
Netcomber 2.0	BIOS Level 123 BIOS date 1/1/96 Mother Board ID XYZ
Device Driver FGHIJ.DRV	BIOS Level 456 BIOS date 2/1/96 Mother Board ID XYZ

Cole, 4:1-24 (EX1002). The metadata for each code update identifies a configuration of a client that is potentially appropriate for the code update. *Id.*, 3:56-4:44 (EX1002). Thus, the association between a code update and its basic system requirements metadata identifies a “known acceptable configuration,” and the database table provides a “list of known acceptable configurations.” Knutson ¶ 76 (EX1006).

Cole discloses that the selection server determines which code updates are potentially appropriate for the client and which code updates are inappropriate for the client based on the client's basic system information. *Cole*, 3:56-4:45 (EX1002). The selection server makes this determination by correlating the metadata in the database to the basic system information obtained from the client. *Id.*, 3:62-65 (EX1002). Referring to the example table provided above, *Cole* discloses that "the BIOS level 123, BIOS date Jan. 1, 1996 and mother board ID XYZ basic system information obtained from [the] client [] is consistent with a device driver file named ABCDE.DRV," and therefore device driver file ABCDE.DRV is potentially appropriate for the client. *Id.*, 3:65-4:30 (EX1002). However, the device driver file named FGHIJ.DRV is inappropriate for the client and is eliminated from further consideration, because its required BIOS level (i.e. BIOS level 456), does not match the BIOS level of the client. *Id.*, 3:65-4:34 (EX1002). Knutson ¶ 77 (EX1006).

By correlating the client's basic system information with information in the database table, the selection server "compares" the basic system information ("information specifying at least one additional component currently implemented in the electronic device"), a code update (a "determined [device] component") and a system requirement identified in the metadata (also a "determined [device]

component”) with the known potentially appropriate configurations in the database table (a “list of known acceptable configurations”). Knutson ¶ 78 (EX1006). Thus, *Cole* discloses “comparing the determined component and information specifying at least one additional component currently implemented in the electronic device” with “a list of known acceptable configurations for the electronic device.” Knutson ¶¶ 73, 78 (EX1006).

As discussed above in Section VII.A.2.c, if a narrow interpretation of a “device component required to implement the reconfiguration request” is taken requiring that the device component be a particular component requested by an electronic device in a reconfiguration request, it would have been obvious to a POSA at the time of the alleged invention to modify *Cole* to incorporate these features in light of the teachings of *MacInnis*. See Section VII.A.2.c, *supra*. Knutson ¶¶ 73, 79 (EX1006).

To the extent the claim is interpreted to require both a “list of known acceptable configurations for the electronic device” and a “list of known unacceptable configurations for the electronic device,”⁵ *Cole* teaches the use of

⁵ During prosecution, the Examiner appears to have interpreted “at least one of a list of known acceptable configurations ... and a list of known unacceptable configurations” as requiring *only one* of a list of known acceptable configurations

both lists. Knutson ¶ 80 (EX1006). The logic of the selection update program⁶ allows the selection server to identify client configurations that are (1) potentially appropriate or (2) inappropriate for code updates. *See Cole*, 3:56-4:45 (EX1002). Knutson ¶ 80 (EX1002). A POSA would recognize that, in executing a stored selection update program that identifies both potentially appropriate (“acceptable”) and inappropriate (“unacceptable”) code updates by referencing a database table, the selection server compares the basic system information (“information specifying at least one additional component currently implemented in the *or* a list of known unacceptable configurations. *See* FH of ’607 application, Notice of Allowance of 07/29/2002 (EX1009) (the Examiner appearing to equate an “acceptable list” with “one of an acceptable and an unacceptable list”). However, should this claim language be interpreted as requiring both a list of known acceptable configurations *and* a list of known unacceptable configurations (*see, e.g., SuperGuide Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870 (Fed. Cir. 2004)), these features were well-known, as discussed herein.

⁶ The ’088 patent explains that a “list” includes “any stored representation of information indicative of component compatibility,” and can be in the form of any “po[r]tion of a program executed by the reconfiguration manager.” ’088 patent, 3:64-4:11 (EX1001); *see also* Section VI.A, *supra*.

electronic device”), a code update (a “determined [device] component”) and a system requirement identified in the metadata (also a “determined [device] component”) with both a “list of known acceptable configurations” and a “list of known unacceptable configurations.” Knutson ¶¶ 73, 80 (EX1006).

Use of both a list of known acceptable configurations and a list of known unacceptable configurations to determine whether software is appropriate for a client was well-known at the time the ’088 patent was filed. Knutson ¶ 81 (EX1006). As another example, *Elgressy* discloses a system that stores two lists, (1) a list of client resources allowed for use by an Executable Object (EO) (e.g., Java Applet, Active-X OCX, Win32 Executable, DLL) and (2) a list of client resources prohibited for use by an EO. *Elgressy*, 1:31-67, 3:8-20, 4:14-32, 5:40-60, 6:25-55 (EX1004). Knutson ¶ 81 (EX1006).

Elgressy discloses that an EO can be downloaded to a computer over a network, and when detected by a gateway, the gateway compares the resources of the computer that the EO needs to utilize against a Security Policy. *Elgressy*, 1:31-67, 3:1-20, 4:14-27, 5:40-60, 6:25-36, FIG. 2 (EX1004). If the resources the EO needs to utilize are “included in the list of the resources allowed for use by the Security Policy,” the EO is allowed to pass through the gateway and reach the computer. *Elgressy*, 3:8-13, 4:14-20, 5:48-54, 6:32-36, FIG. 2 (EX1004). But if

the resources the EO needs to utilize are “included in the list of the resources prohibited for use by the Security Policy,” the EO is prevented from reaching the computer. *Elgressy*, 3:14-20, 4:20-27, 5:55-60, 6:25-32, FIG. 2 (EX1004). Knutson ¶ 82 (EX1006).

Cole and *Elgressy* are each directed to systems that determine whether software is acceptable or unacceptable for a device. Knutson ¶ 83 (EX1006). In light of the teachings of *Elgressy*, it would have been obvious to a POSA at the time of the alleged invention to modify *Cole* so that the selection server utilizes both a table listing known appropriate configurations for code updates and a table listing known inappropriate configurations for code updates. *Id.*

A POSA would have been motivated to make this modification to provide greater versatility in handling an upgrade request. Knutson ¶ 84 (EX1006). For example, *Cole* discloses that when the basic system information indicates that a client has BIOS level 123 and BIOS date Jan. 1, 1996, device driver FGHIJ.DRV is eliminated from further consideration since it is inappropriate for the client. *Cole*, 4:25-34 (EX1002). However, a POSA would have recognized that device driver FGHIJ.DRV might be made appropriate if the client were to first update its BIOS to level 456 with date 2/1/96. *See Cole*, 4:10-34 (EX1002); *see also Pitzel*, 9:66-10:12 (disclosing that a requested component could be made compatible with

a device by providing “additional components that are necessary for proper operation” of the requested component) (EX1005). Modifying *Cole* to provide both a list of known appropriate configurations *and* a list of known inappropriate configurations, such as that taught by *Elgressy*, would have allowed for the elimination of code updates that cannot be made appropriate for a client, while providing for the handling of situations where a currently inappropriate code update might be made appropriate for the client (e.g., with a simple intermediary update on a client). Knutson ¶¶ 73, 84 (EX1006).

Additionally, a POSA would have been motivated to add a second table for identifying inappropriate code updates for a client to address security concerns associated with downloading software to the client. *See Elgressy*, 1:41-2:16 (EX1005). For example, even if a code update is operable on a client, it may be desirable to prevent the downloading of that code update for any number of reasons. *See, e.g., Elgressy*, 1:56-59 (to prevent “jam[ming] the network,” “irreversible damage to the local database, workstations and servers,” and “unauthorized retrieval of information from the servers/workstations”). For this reason as well, a POSA would have been motivated to modify *Cole* to include a second table for identifying code updates that are inappropriate for the client for reasons other than incompatibility. Knutson ¶¶ 73, 85 (EX1006).

Modifying *Cole* to utilize tables that identify both known appropriate configurations and known inappropriate configurations would have been well within the abilities of a POSA and could be easily accomplished with a high chance of success. Knutson ¶ 86 (EX1006). Only software techniques that were conventional at the time of the alleged invention would have been necessary to make such a modification. Knutson ¶¶ 73, 86 (EX1006).

e) “generat[[ing]/[e]] information indicative of an approval or a denial of the reconfiguration request based at least in part on the [[result of the comparing step]/[comparison operation]]”

Cole discloses that, for each code update that is potentially appropriate for a client, the selection server sends FTP addressing information of a corresponding recognizer program to the client. *Cole*, 3:56-4:51 (EX1002). Because the FTP addressing information is sent when the selection server has determined that the code update is potentially appropriate for the client, the “FTP addressing information” is information generated by the selection server that is “indicative of an approval ... of the reconfiguration request” based at least in part on the result of the comparing step. Knutson ¶ 87 (EX1006).

Cole also discloses that, when a code update is inappropriate for a client, the code update is “eliminated from further consideration.” *Cole*, 4:30-34 (EX1002); *see also id.* at 3:57-62 (EX1002). Thus, in eliminating a code update from consideration, the selection server generates “information indicative of ... a denial

of the reconfiguration request” based at least in part on the result of the comparing step. Knutson ¶ 88 (EX1006),

Cole further discloses that appropriate code updates are included in a list, from which the user can select which code updates to download. *Cole*, 5:56-6:25 (EX1002). This list is also information generated by the selection server that is “indicative of an approval ... of the reconfiguration request based at least in part on the result of the comparing step.” Knutson ¶ 89 (EX1006).

As discussed above in Section VII.A.2.c, it would have been obvious to a POSA to modify *Cole* so that the selection server receives a request for a particular code update, and determines whether this code update is appropriate for the client, as taught by *MacInnis*. Knutson ¶ 90 (EX1006). As further discussed above in Section VII.A.2.d, it would also have been obvious to a POSA to modify *Cole* to include both a list of appropriate configurations for code updates and a list of inappropriate configurations for code updates, and to approve or deny the downloading of a code update to a client based on comparison with both of these lists, as taught by *Elgressy*. *Id.*

f) Claim 11 - “a memory for storing at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device”

Claim 11 also requires that the “at least one of a list of known acceptable

configurations for the electronic device and a list of known unacceptable configurations for the electronic device” be stored in a memory. As discussed above in Section VII.A.2.a, *Cole* discloses a selection server that is a “computer workstation” with “associated programming” and that utilizes a database with a “list of known acceptable configurations for the electronic device.” At the time of the alleged invention, a POSA would have recognized that a computer workstation with programming and a database must necessarily include memory. Knutson ¶ 91 (EX1006).

As further discussed above in Section VII.A.2.d, *Cole*, alone or in combination with *Elgressy*, teaches a list of acceptable configurations and a list of known unacceptable configurations. *Cole* discloses that the selection server includes a database, which stores the metadata for the code updates. *See Cole*, 2:66-3:6, 3:56-4:44, FIG. 2 (EX1002). Knutson ¶ 92 (EX1006).

g) Claim 11 - “a processor coupled to the memory and operative” to perform the steps of sections VII.A.2.a-e above

Claim 11 further requires a “processor coupled to the memory” which is operative to perform the steps of sections VII.A.2.a-e that correspond to claim 11. As discussed above in Sections VII.A.2.a-e, *Cole* discloses that the selection server is a “computer workstation” with “associated programming” that can access a database to determine whether a code update is potentially appropriate for a client.

Knutson ¶ 93 (EX1006). A POSA would recognize that the selection server necessarily includes a “processor coupled to the memory” storing the program for implementing the steps of Sections VII.A.2.a-e. *Id.*

3. Claims 2⁷ and 12 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “generat[[ing]/[e]] information indicative of an approval of the reconfiguration request if the determined[-]component and the additional component are consistent with a given one of the known acceptable configurations”

As discussed above, *Cole*, alone or in combination with *MacInnis*, teaches generating “information indicative of an approval ... of the reconfiguration” (e.g., sending FTP addressing information, sending a list of appropriate code updates) based at least in part on the result of the comparing step. *See* Sections VII.A.2.c-e, *supra*. Thus, *Cole*, alone or in combination with *MacInnis*, teaches generating “information indicative of an approval of the reconfiguration request if the determined[-]component and the additional component are consistent with a given one of the known acceptable configurations.” Knutson ¶¶ 96, 97 (EX1006).

⁷ Although this Petition shows that *Cole*, alone or in combination with *MacInnis*, teaches the subject matter of claim 2, such a showing is not required since claim 2 is a method claim whose single step is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

4. Claims 3⁸ and 13 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “download[ing] the determined component to the electronic device if the determined component and the additional component are consistent with a given one of the known acceptable configurations”

As discussed above, *Cole*, alone or in combination with *MacInnis*, teaches that the selection server identifies whether the “determined component and the additional component are consistent with a given one of the known acceptable configurations.” See Sections VII.A.2.c-e, and VII.A.3, *supra*. Knutson ¶ 101 (EX1006).

Cole discloses that, for each code update that is potentially appropriate for a client, the selection server sends FTP addressing information of a corresponding recognizer program. *Cole*, 4:35-41 (EX1002). This recognizer program “further investigates the hardware, software and other components of the client [] for information to assist the server in determining whether the corresponding update is appropriate for the client.” *Id.*, 5:2-6 (EX1002). If a code update is still deemed

⁸Although this Petition shows that *Cole*, alone or in combination with *MacInnis*, teaches the subject matter of claim 3, such a showing is not required since claim 3 is a method claim whose single step is conditional. See *Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

appropriate for the client, the client can download the code update. *Id.*, 5:56-6:49 (EX1002). Knutson ¶¶ 100, 102 (EX1006).

5. Claims 4 and 14 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “compar[[ing]/[e]] the determined[,] component and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device”

As discussed above in Section VII.A.2.d, *Cole*, alone or in combination with *MacInnis* and/or *Elgressy*, teaches comparing the determined component “and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device.” Knutson ¶¶ 105, 106 (EX1006).

b) “generat[[ing]/[e]] information indicative of a denial of the reconfiguration request if the determined component and the additional component are consistent with a given one of the known unacceptable configurations”⁹

⁹ Although this Petition shows that *Cole*, alone or in combination with *MacInnis* and/or *Elgressy*, teaches this limitation of claim 4, such a showing is not required since claim 4 is a method claim and this limitation recites a single step that is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

As discussed above, the selection server includes a selection update program with logic that identifies both known potentially appropriate client configurations (“known acceptable configurations”) and known inappropriate client configurations (“known unacceptable configurations”) for the code updates. *See* Section VII.A.2.d, *supra*. Knutson ¶ 108 (EX1006). If the basic system information sent from a client does not match the metadata associated with a code update, the code update is inappropriate for the client and is eliminated from further consideration. *See* Section VII.A.2.d, *supra*. This elimination from further consideration is “information indicative of a denial of the reconfiguration request if the determined component and the additional component are consistent with a given one of the known unacceptable configurations.” Knutson ¶¶ 107, 108 (EX1006).

Moreover, as discussed above in Section VII.A.2.d, *Elgressy* discloses preventing an EO from passing to a client when the computer resources the EO requires are included in a list of resources prohibited for use on that client. Knutson ¶ 109 (EX1006). As further discussed above, it would have been obvious to a POSA to modify *Cole* to utilize a table that identifies known inappropriate configurations, such as taught by *Elgressy*. Knutson ¶¶ 107, 109 (EX1006).

6. Claims 5 and 15 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “*compar[[ing]/[e]] the determined component and information specifying*

at least one additional component currently implemented in the electronic device with [[the]/[a]] list of known unacceptable configurations for the electronic device”

As discussed above *Cole*, alone or in combination with *MacInnis* and/or *Elgressy*, teaches comparing the determined component “and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device.” See Sections VII.A.2.d, VII.A.5.a, *supra*. Knutson ¶¶ 112, 113 (EX1006).

b) “generat[[ing]/[e]] information indicating that the requested reconfiguration is unknown if the determined component and the additional component are not consistent with a given one of the known acceptable or unacceptable configurations”¹⁰

As discussed above in Sections VII.A.2.d, *Cole*, in combination with *MacInnis* and/or *Elgressy*, teaches utilizing a list of system characteristics known

¹⁰ Although this Petition shows that *Cole*, alone or in combination with *MacInnis* and/or *Elgressy*, teaches this limitation of claim 5, such a showing is not required since claim 5 is a method claim and this limitation recites a single step that is conditional. See *Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

to be potentially appropriate for code updates, and storing and utilizing a list of system characteristics known to be inappropriate for code updates, to determine whether a code update can be downloaded to a client. Knutson ¶ 115 (EX1006). Any basic system information of a client that does not match the system characteristics for a code update in either of these lists would be basic system information for which the appropriateness of the code update is unknown. *Id.* It would have been obvious to a POSA at the time of the alleged invention to generate information indicating that a requested reconfiguration is unknown when the basic system information of the client does not correspond to the system characteristics for a code update in lists of known potentially appropriate and inappropriate configurations. *Id.* Indeed, identifying an unknown condition in computer software is a practice computer programmers used for many years prior to the filing date of the '088 patent. *Id.* A POSA would have been motivated to make such a modification to make software more robust and less error-prone, and to address situations not matching the known situations the software is programmed to handle. Knutson ¶¶ 114, 115 (EX1006).

7. Claims 6 and 16 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “transmit[ting] in response to the reconfiguration request a list of additional components required in the electronic device in order to implement the reconfiguration [request]”

Cole discloses that, after the client “contacts” the selection server to begin a

session and sends the current level of the update manager, the scout, the service application, and the download routine within the client to the selection server, the selection server determines whether the client's update manager, scout, service application, and download routine are the latest versions. *Cole*, 3:14-27 (EX1002). If they are not, the selection server sends FTP addressing information to the client to allow the client to retrieve the latest versions from the content server. *Id.*, 3:28-31 (EX1002). The selection server also sends the client FTP addressing information to allow the client to retrieve "a basic system information recognizer program [] stored in [the] content server." *Id.*, 3:32-45 (EX1002). Thus, in providing FTP addressing information for the update manager, scout, service application, download routine, and basic system recognizer, the selection server provides "a list of additional components required in the electronic device in order to implement the reconfiguration" in response to the "contact" ("reconfiguration request") from the client or the information about the level of the update manager, scout, service application, and download routine ("reconfiguration request") from the client. Knutson ¶¶ 118, 119 (EX1006).

Cole also discloses that, after the basic system information is sent from the client to the selection server (*Cole*, 3:52-55 (EX1002)), the selection server sends FTP addressing information for the client to retrieve recognizer programs

corresponding to the potentially appropriate code updates (*Cole*, 4:35-41 (EX1002)). Thus, in providing FTP addressing information for the recognizer programs corresponding to the potentially appropriate code updates, the selection server also provides “a list of additional components required in the electronic device in order to implement the reconfiguration” that are transmitted in response to the “basic system information” (“reconfiguration request”) from the client. Knutson ¶¶ 118, 120 (EX1006).

8. Claims 7 and 17 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “wherein the information specifying at least one additional component currently implemented in the electronic device includes identifiers of each of the components in a set of components currently implemented in the electronic device”

As discussed above, *Cole* discloses that the client sends “basic system information” to the selection server. *See* Section VII.A.2.d, *supra*. The basic system information can include, for example, the client’s system model, pre-load software level, BIOS level, type of operating system, or mother board ID. *See id.* *Cole* provides an example where the basic system information of the client indicates that a BIOS level of “123,” a BIOS date of “Jan. 1, 1996,” and a mother board ID of “XYZ.” *Cole*, 3:50-52 (EX1002). Each of these (e.g., BIOS level “123,” BIOS date “Jan. 1, 1996,” mother board ID “XYZ”) is an “identifier” of a component (e.g., BIOS, mother board) in a set of components currently

implemented in the client. Knutson ¶¶ 123, 124 (EX1006).

9. Claims 8 and 18 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “wherein the identifiers of each of the components in the set of components are included in the reconfiguration request [transmitted by the electronic device]”

As discussed above, the “basic system information” sent from the client to the selection server is a “reconfiguration request.” See Section VII.A.2.b, *supra*. As also discussed above, the basic information can include “identifiers of each of the components in a set of components implemented in the electronic device.” See Section VII.8.a, *supra*. Thus, *Cole* discloses that the identifiers (e.g., “123,” “Jan. 1, 1996,” “XYZ”) of each of the components (e.g., BIOS, mother board) in the set of components are included in the basic system information (the “reconfiguration request”) sent to the selection server. Knutson ¶¶ 127, 128 (EX1006).

10. Claims 9 and 19 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) “wherein the reconfiguration request comprises a request for an upgrade of at least one of a software component and a hardware component of the electronic device”

As discussed above, each of (i) a “contact” at the selection server from the client, (ii) information about a level of certain applications in the client, such as an update manager, a scout, a service application, and a download routine, and (iii) “basic system information” sent from the client to the selection server, is a “reconfiguration request.” See Section VII.A.2.b, *supra*. Knutson ¶ 132

(EX1006). Each of these requests is also a request for an upgrade of a software component of the electronic device. Knutson ¶¶ 131, 132 (EX1006). For example, after the client contacts the selection server and sends a level of the update manager, the scout, the service application, and the download routine, the selection server provides FTP addressing information to newer versions of the update manager, the scout, the service application, and the download routine if they are outdated. *Cole*, 3:14-39 (EX1002). Knutson ¶¶ 131, 132 (EX1006).

Cole also discloses that the selection server determines which code updates are appropriate for the client. *Cole*, 3:56-62 (EX1002). These code updates can be updates to software already stored on the client. *Id.*, 5:13-18 (“Recognizer program [] determines ... if the size of file ABCDE.DRV. stored by the client [] is the same as the size of file ABCDE.DRV. stored in the data base []. If so, then [the] client [] has the latest update. If not, then [the] client [] likely has an old version which could be updated.”) (EX1002). Thus, each of (i) the “contact,” (ii) the information about the level of the update manager, the scout, the service application, and the download routine sent from the client to the selection server, and (iii) the basic system information sent to the selection server is a “request for an upgrade of ... a software component ... of the electronic device.” Knutson ¶¶ 131, 133 (EX1006).

A POSA would recognize that the installation of new software in the memory of a computer necessarily involves a modification (“upgrade”) of that memory. Knutson ¶ 134 (EX1006). Thus, each of (i) the “contact,” (ii) the information about the level of the update manager, the scout, the service application, and the download routine sent from the client to the selection server, and (iii) the basic system information sent to the selection server is also a “request for an upgrade of ... a hardware component of the electronic device.” Knutson ¶¶ 131, 134 (EX1006).

11. Claims 10 and 20 are obvious in view of *Cole*, *MacInnis*, and *Elgressy*

a) *“wherein the reconfiguration request is received from the electronic device over a network connection established [[with a reconfiguration manager implementing the receiving, determining, comparing and generating steps]/[, with a reconfiguration manager which includes the memory and processor]]”*

As discussed above, *Cole* discloses a selection server (“reconfiguration manager”) that receives a “reconfiguration request” from the client (“electronic device”). See Section VII.2.A.b, *supra*. As further discussed above, *Cole*, alone or in combination with *MacInnis* and/or *Elgressy*, teaches that the selection server includes a memory and processor and implements the “receiving,” “determining,” “comparing,” and “generating” steps. See Section VII.2.A, *supra*. Knutson ¶¶ 137, 138 (EX1006).

B. Ground II: Claims 1-21 are unpatentable as being obvious in view of *Pitzel, Cole, and Elgressy*

Pitzel, Cole, and Elgressy were not considered during prosecution, but are highly relevant to the claims of the '088 patent.

1. Overview of *Pitzel*

As depicted in the figure below, *Pitzel* discloses a system that includes a component server (outlined in red), a network (outlined in blue), and a client (outlined in green).

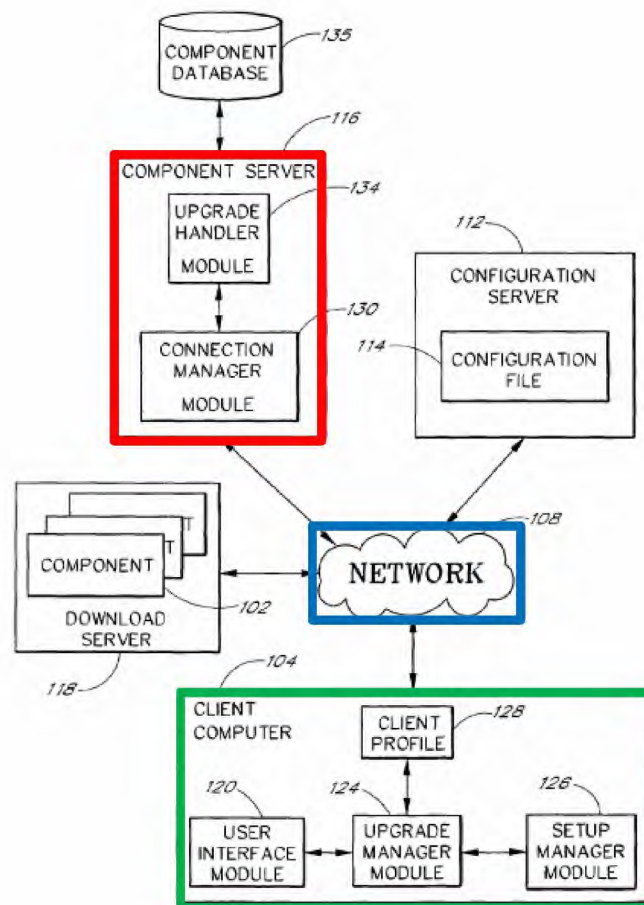


FIG. 1

Pitzel, FIG. 1 (EX1005).

The system is configured to install one or more components on the client. *Pitzel*, 3:13-14 (EX1005). Knutson ¶ 140 (EX1006).

Pitzel discloses that a client “generates an upgrade request which requests the location of one or more components.” *Pitzel*, 7:36-44 (EX1005). This upgrade request is sent to the component server, and includes information about the requested components and “client conditions.” *Id.*, 6:8-17, 7:36-44, 9:23-28 (EX1005). The client conditions can include, for example, “a preferred operating language,” “the name of the operating system of the client computer,” “any version number that may be associated with the operating system,” “the existence of one or more other components of the client computer,” a “user identification number,” “product information,” “distribution codes,” or an “identifier that is associated with the microprocessor.” *Pitzel*, 4:43-54, 6:40-7:11, 8:65-9:21 (EX1005). Knutson ¶ 141 (EX1006).

The component server “analyzes the upgrade request” by “us[ing] the client conditions ... to determine an appropriate version of the component which was requested by the user.” *Pitzel*, 7:49-56 (EX1005). An “appropriate version” of a component is a version that is compatible with the client conditions. *See Pitzel*, 6:41-44, 9:57-65 (EX1005). After analyzing the upgrade request, the component server “generates an upgrade response message” which “identifies the location of

the components requested by the client,” if appropriate versions can be located. *Pitzel*, 7:49-59, 10:13-25 (EX1005). The client then “analyzes the upgrade response message to determine the location of requested components” and “downloads and installs the requested components.” *Id.*, 7:60-67 (EX1005). Knutson ¶ 142 (EX1006).

2. Claims 1, 11, and 21 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) Claims 1, 11 - “[a] processor-implemented method”/“[a]n apparatus” for controlling the reconfiguration of an electronic device”

Claim 21 - “[a]n article of manufacture comprising a machine-readable medium containing one or more software programs”

Pitzel discloses “a system and a method for updating components in a client” that includes a component server. *Pitzel*, Abstract (EX1005). The component server includes a processor. *Id.*, 3:46-63 (EX1005). The component server may use a particular type of operating system (*Id.*, 3:58-63 (EX1005)), and may include software modules for carrying out the processes within the component server (*Id.*, 4:8-30, 5:47-63 (EX1005)). Thus, the component server is an apparatus, and comprises a processor that executes a stored software program to implement a method. Knutson ¶ 147 (EX1006).

The component server receives an upgrade request from the client (“electronic device”) “which requests the location of one or more components” for download and installation on the client. *Pitzel*, 7:36-41, 7:60-67, 8:1-9 (EX1005).

The one or more components can be “any item of executable code or data, or group of such items, that is capable of being processed by a computer.” *Id.*, 3:13-17 (EX1005). Knutson ¶ 148 (EX1006).

With the upgrade request, the component server also receives client conditions, which can be in the form of a client profile. *Pitzel*, 9:6-28 (EX1005). The component server “uses the client conditions to select components that are compatibly operable with the client.” *Id.*, 9:57-65 (EX1005). If the component server can identify versions of the components that are compatible with the client, an upgrade response message is sent to the client identifying the locations of computers hosting the requested components. *Id.*, 10:21-25 (EX1005). The client can then download the components (*Id.*, 10:50-58 (EX1005)) and install the components (*Id.*, 11:20-31 (EX1005)). If the component server cannot identify versions of the components that are compatible with the client, the upgrade request is denied and the user is informed that the upgrade failed. *Id.*, 10:13-20, 10:39-49 (EX1005). Knutson ¶ 149 (EX1006).

Thus, in determining whether components compatible with a client exist, and either providing the location of the components or denying the upgrade request based on that determination, the component server “control[s] the reconfiguration of an electronic device.” Knutson ¶¶ 146, 150 (EX1006).

b) “receiv[ing]/[e] information representative of a reconfiguration request relating to [[the]/[[an]] electronic device”

Pitzel discloses that the client sends an upgrade request to the component server (*Pitzel*, 9:23-25 (EX1005)), which “requests the location of one or more components” (*Pitzel*, 7:36-38 (EX1005)). Knutson ¶¶ 151, 152 (EX1006).

c) “determin[ing]/[e] at least one device component required to implement the reconfiguration request”

Pitzel discloses that the upgrade request identifies the requested component. *Pitzel*, 8:40-44, 8:47-60, 9:23-25 (EX1005). The component server then “determine[s] an appropriate version of the [requested] component.” *Id.*, 7:53-56 (EX1005). The component server determines the requested component when it “analyzes the upgrade request.” *Id.* Thus, the component server determines at least one device component (i.e., a component requested by the user) required to implement the reconfiguration request. Knutson ¶¶ 153, 154 (EX1006).

Pitzel also discloses that the component server can “determine one or more additional components that are necessary for proper operation in addition to those components [] requested by the client computer.” *Pitzel*, 9:66-10:2 (EX1005). For example, “one of the requested components may be dependent on the existence of another component in the client computer [] for proper operation.” *Id.*, 10:2-5 (EX1005). In determining any additional component necessary for proper

operation, the component server is also determining at least one device component (i.e., an additional component required for proper operation) required to implement the reconfiguration request. Knutson ¶¶ 153, 155 (EX1006).

Pitzel further discloses that a client profile including client conditions is sent from the client to the component server as part of the upgrade request. *Pitzel*, 7:36-44, 9:23-28 (EX1005). The client conditions can include information related to the software or hardware components on the client computer, such as the name of the operating system of the client computer, a version number associated with the operating system, the existence of one or more other components of the client computer, product information, distribution codes, and a unique identifier associated with a microprocessor. *See Pitzel*, 4:41-52, 6:40-7:11, 8:64-9:22, 1:43-47, 2:3-5 (EX1005). The component server uses the client conditions to determine a compatible version of the requested component. *Pitzel*, 7:49-56, 9:57-65 (EX1005). These client conditions are also “device component[s]” required to implement the upgrade request (“required to implement the reconfiguration request”), because the components of the upgrade are selected on the basis of these client conditions. Knutson ¶ 156 (EX1006). The component server determines the client conditions when it “analyzes the upgrade request.” *See Pitzel*, 7:49-56 (EX1005), Knutson ¶¶ 153, 156 (EX1006).

In sum, *Pitzel* discloses that the component server determines information about (i) a requested component, (ii) an additional component necessary for proper operation, and (iii) a component already on the client, each of which is a “device component required to implement the reconfiguration request.” Knutson ¶¶ 153, 157 (EX1006).

d) “compar[[ing]/[e]] the determined component and information specifying at least one additional component currently implemented in the electronic device with at least one of [[a]/[the]] list of known acceptable configurations for the electronic device and [[a]/[the]] list of known unacceptable configurations for the electronic device”

Pitzel discloses that the component server uses the client conditions provided to determine an appropriate version of the requested component. *Pitzel*, 7:53-56 (EX1005). The component server “select[s] components that are compatibly operable with the client.” *Id.*, 6:41-44, 9:57-65 (EX1005). Thus, in identifying a version of the requested component that is compatible with the client, the component server is comparing the requested component (the “determined component”) and the client conditions (“information specifying at least one additional component currently implemented in the electronic device”) with information about known compatible or incompatible configurations (“at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the device”). Knutson ¶¶ 158, 159 (EX1006).

As discussed above in Section VII.B.2.c, an additional component necessary for proper operation is also a determined “device component.” *Pitzel* discloses that “dependency information” can be stored in the database, so that when a version of a requested component to be installed on the client requires an additional component for proper operation, and that additional component is missing, the additional component can be identified and provided to the client. *Pitzel*, 9:66-10:12 (EX1005). Thus, in identifying an additional component required to make the requested component compatible with the client, the component server is comparing the additional component (the “determined component”) and the client conditions (“information specifying at least one additional component currently implemented in the electronic device”) with information about known compatible or incompatible configurations (“at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the device”). Knutson ¶¶ 158, 160 (EX1006).

As also discussed above in Section VII.B.2.c, a component identified by the client conditions, as disclosed by *Pitzel*, is also a determined “device component.” *Pitzel* discloses that multiple components on the client can be identified in the “client conditions,” such as a name or version number of the operating system, one or more other components of the client computer, product information, distribution

IPR2019-00056
Patent 6,467,088

codes, and a unique identifier associated with a microprocessor. *Pitzel*, 4:41-52, 6:40-7:11, 8:64-9:22, 1:43-47, 2:3-5 (EX1005). Each of these is both a “device component required to implement the reconfiguration request” and an “additional component currently implemented in the electronic device.” Thus, in identifying a version of the requested component that is compatible with the client, the component server compares one of the components identified in the client conditions (the “determined component”) and another of the components identified in the client conditions (“information specifying at least one additional component currently implemented in the electronic device”) with information about known compatible or incompatible configurations (“at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the device”). Knutson ¶¶ 158, 161 (EX1006).

Patent Owner may argue that *Pitzel* does not disclose a “list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device” and thus does not compare components with such a list. Petitioner disagrees.¹¹ However, database tables and

¹¹ The '088 patent explains that “[t]he term ‘list’ as used herein is [] intended to include any stored representation indicative of component compatibility.” '088 patent, 4:6-8 (EX1001).

other such lists of acceptable and unacceptable configurations were well-known at the time of the alleged invention. Knutson ¶ 162 (EX1006).

For example, as discussed above in Section VII.A, *Cole* discloses a “selection server” that utilizes a database table to identify potentially appropriate code updates for a client. *Cole*, Abstract, 2:52-53 (EX1002). Like *Pitzel*, *Cole* discloses that its server receives an update request, and information about characteristics of the client, which in *Cole* is called “basic system information.” *See Cole*, 3:14-55 (EX1002). Knutson ¶ 163 (EX1006). The basic system information can include, for example, information about the client’s system model, pre-load software level, BIOS level, type of operating system, and mother board ID. *Cole*, 3:47-55, 4:10-30 (EX1002). Like *Pitzel*, the server of *Cole* determines which code updates are appropriate for a client and which code updates are inappropriate for the client. *See Cole*, 1:45-65, 3:57-4:45 (EX1002). Knutson ¶ 163 (EX1006).

In light of the teachings of *Cole*, it would have been obvious to a POSA at the time of the alleged invention to modify *Pitzel* so that, in determining whether particular versions of requested components are compatible with the client conditions of the client, the component server would compare client conditions and requested components with information in a database table identifying known

potentially appropriate configurations. Knutson ¶ 164 (EX1006). Given the similarities in structure, objectives, and operation between *Pitzel* and *Cole*, a POSA would have been motivated to utilize a database table identifying software updates and their corresponding system requirements, such as taught by *Cole*, as an efficient way to organize and identify records (i.e., the compatibility requirements of pieces of software). Knutson ¶ 164 (EX1006). Indeed, at the time of the alleged invention, database lookup tables were well-known techniques for efficiently locating information. *Id.*

Modifying *Pitzel* to utilize a database table for locating information (e.g., versions of components that are compatible with the conditions of a client), as was conventionally done at the time of the alleged invention, would have been well within the abilities of a POSA and could be easily accomplished with a high chance of success. Knutson ¶ 165 (EX1006). Only software techniques that were conventional at the time of the alleged invention would have been necessary to provide for identifying a version of a component and its required client conditions in a database table. Knutson ¶¶ 158, 165 (EX1006).

To the extent the claim is interpreted to require both a “list of known acceptable configurations for the electronic device” and a “list of known

unacceptable configurations for the electronic device,”¹² *Pitzel*, alone or in combination with *Cole*, teaches the use of both lists. Knutson ¶ 166 (EX1006). The logic of the component server’s upgrade handler module program¹³ described in *Pitzel* allows the component server to identify client configurations that are (1) compatible or (2) incompatible with versions of components. *See Pitzel*, 4:10-30, 5:47-63, 9:59-65 (EX1005). Knutson ¶ 166 (EX1002). A POSA would recognize that in executing a stored upgrade handler module program that identifies both compatible (“acceptable”) and incompatible (“unacceptable”) versions of components, the component server compares the client conditions (each of which is “information specifying at least one additional component currently implemented in the electronic device” or a “device component”), and a requested component (a “determined [device] component”) with both a “list of known acceptable configurations for the electronic device” and a “list of known unacceptable configurations.” Knutson ¶¶ 158, 166 (EX1006).

Using both a list of known acceptable configurations and a list of known unacceptable configurations to determine whether software is appropriate for a client was well-known at the time the ’088 patent was filed. Knutson ¶ 167

¹² *See* footnote 5, *supra*.

¹³ *See* footnote 6, *supra*.

(EX1006). As another example, *Elgressy* discloses a system that stores two lists, (1) a list of client resources allowed for use by an Executable Object (EO) (e.g., Java Applet, Active-X OCX, Win32 Executable, DLL) and (2) a list of client resources prohibited for use by an EO. *Elgressy*, 1:31-67, 3:8-20, 4:14-32, 5:40-60, 6:25-55 (EX1004). Knutson ¶ 167 (EX1006).

Elgressy discloses that an EO can be downloaded to a computer over a network, and when detected by a gateway, the gateway compares the resources of the computer that the EO needs to utilize against a Security Policy. *Elgressy*, 1:31-67, 3:1-20, 4:14-27,, 5:40-60, 6:25-36, FIG. 2 (EX1004). If the resources the EO needs to utilize are “included in the list of the resources allowed for use by the Security Policy,” the EO is allowed to pass through the gateway and reach the computer. *Id.*, 3:8-13, 4:14-20, 5:48-54, 6:32-36, FIG. 2 (EX1004). By contrast, if the resources the EO needs to utilize are “included in the list of the resources prohibited for use by the Security Policy,” the EO is prevented from passing through the gateway, and is thereby prevented from reaching the computer. *Id.*, 3:14-20, 4:20-27, 5:55-60, 6:25-32, FIG. 2 (EX1004). Knutson ¶ 168 (EX1006).

Pitzel, *Cole*, and *Elgressy* are each directed to systems that determine whether software is acceptable or unacceptable for a client. Knutson ¶ 169 (EX1006). In light of the teachings of *Elgressy*, it would have been obvious to a

POSA at the time of the alleged invention to modify the combination of *Pitzel* and *Cole* so that the component server utilizes both a table listing known potentially appropriate configurations for requested components and a table listing known inappropriate configurations for requested components. *Id.* A POSA would have been motivated to make this modification to provide greater versatility in handling an upgrade request. *Id.*

For example, *Pitzel* discloses that a requested component might be made compatible with a device by providing “additional components that are necessary for proper operation” of the requested component. *Pitzel*, 9:66-10:12 (EX1005). Modifying the combination of *Pitzel* and *Cole* to provide both a list of known appropriate configurations *and* a list of known inappropriate configurations, such as that taught by *Elgressy*, would have allowed for the elimination of components that cannot be made compatible with the client, while providing for the handling of situations where a component might be made appropriate for the client (e.g., with a simple intermediary update). Knutson ¶¶ 158, 170 (EX1006).

Additionally, a POSA would have been motivated to make this modification to address security concerns associated with downloading software to a client. *See Elgressy*, 1:41-2:16 (EX1005). Knutson ¶ 171 (EX1006). For example, even if a component is compatible with a client, it may be desirable to prevent the

downloading of that component for any number of reasons. *See, e.g., Elgressy*, 1:56-69 (to prevent “jam[ming] the network,” “irreversible damage to the local database, workstations and servers,” and “unauthorized retrieval of information from the servers/workstations”). Knutson ¶ 171 (EX1006). For this reason as well, a POSA would have been motivated to modify the combination of *Pitzel* and *Cole* to include a second table listing components that, despite being operable on a client, are inappropriate for other reasons. Knutson ¶¶ 158, 171 (EX1006).

Modifying the combination of *Pitzel* and *Cole* to include both a list of compatible configurations and a list of incompatible or inappropriate configurations would have been well within the abilities of a POSA and could be easily accomplished with a high chance of success. Knutson ¶ 172 (EX1006). Only software techniques that were conventional at the time of the alleged invention would have been necessary to make such a modification. Knutson ¶¶ 158, 172 (EX1006).

e) “generat[[ing]/[e]] information indicative of an approval or a denial of the reconfiguration request based at least in part on the [[result of the comparing step]/[comparison operation]]”

Pitzel discloses that the component server determines whether locations of versions of requested components compatible with the client computer can be identified. *See Pitzel*, 10:13-17 (EX1005). If they cannot, the upgrade request is denied. *See Pitzel*, 10:17-20 (EX1005). Thus, if compatible versions of the

requested components cannot be located, the component server denies the upgrade request (“generates information indicative of a denial of the reconfiguration request”). Knutson ¶¶ 173, 174 (EX1006).

If the component server can identify the locations of compatible versions of the requested components, the component server “creates an upgrade response message to the client computer” that “identif[ies] the location of the computers that are hosting the requested components.” *Pitzel*, 10:21-25 (EX1005). Thus, if the compatible versions of the requested components can be identified, the component server creates an upgrade response message that identifies the location of the computers that are hosting the requested components (“generates information indicative of an approval of the reconfiguration request”). Knutson ¶¶ 173, 175 (EX1006).

Moreover, as discussed above in Section VII.B.2.d, modifying *Pitzel* to utilize a database table storing information about known appropriate configurations and to generate information indicating whether a version of a component is compatible with the conditions of a client based on a comparison with the table, as taught by *Cole* and *Elgressy*, would have been obvious to a POSA at the time of the alleged invention. Knutson ¶¶ 173, 176 (EX1006).

f) Claim 11 - “a memory for storing at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device”

Claim 11 also requires that the “at least one of a list of known acceptable configurations for the electronic device and a list of known unacceptable configurations for the electronic device” be stored in a memory. As discussed above, *Pitzel* discloses a component server that includes a processor and uses an operating system and an upgrade handler module for identifying components to be downloaded to a client based upon the contents of a client profile. *See Pitzel*, 3:46-63, 5:47-63 (EX1005); Sections VII.B.2.a-e, *supra*. At the time of the alleged invention, a POSA would have recognized that a server with software (e.g., an operating system, an upgrade handler module) must necessarily include memory. *Knutson* ¶ 177 (EX1006).

As further discussed above in Section VII.B.2.d, *Pitzel*, alone or in combination with *Cole* and/or *Elgressy*, teaches a list of known acceptable configurations and a list of known unacceptable configurations. *Pitzel* discloses that the upgrade handler module of the component server “identifies components to be downloaded to the client computer [] based upon the contents of the client profile.” *See Pitzel*, 5:47-63 (EX1005). *Knutson* ¶ 178 (EX1006).

g) Claim 11 - “a processor coupled to the memory and operative” to perform the steps of sections VII.A.2.a-e above

Claim 11 further requires a “processor coupled to the memory” which is operative to perform the steps of sections VII.B.2.a-e that correspond to claim 11. As discussed above in Sections VII.B.2.a-e, *Pitzel* discloses a component server that can determine whether versions of requested components are compatible with the client conditions of a client computer. *Pitzel* further discloses that the component server has a processor for performing this determination. *See Pitzel*, 3:46-63 (EX1005). Thus, *Pitzel* discloses a “processor coupled to the memory.” Knutson ¶ 179 (EX1006).

3. Claims 2¹⁴ and 12 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “generat[[ing]/[e]] information indicative of an approval of the reconfiguration request if the determined[-]component and the additional component are consistent with a given one of the known acceptable configurations”

As discussed above, *Pitzel*, alone or in combination with *Cole*, teaches “generating information indicative of an approval ... of the reconfiguration

¹⁴ Although this Petition shows that *Pitzel*, alone or in combination with *Cole*, teaches the subject matter of claim 2, such a showing is not required since claim 2 is a method claim whose single step is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

request” based at least in part on the result of the comparing step. *See* Sections VII.B.2.d and VII.B.2.e, *supra*. Thus, *Pitzel*, alone or in combination with *Cole*, teaches generating “information indicative of an approval of the reconfiguration request if the determined[-]component and the additional component are consistent with a given one of the known acceptable configurations.” Knutson ¶¶ 182, 183 (EX1006).

4. Claims 3¹⁵ and 13 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “download[ing] the determined component to the electronic device if the determined component and the additional component are consistent with a given one of the known acceptable configurations”

As discussed above, *Pitzel*, alone or in combination with *Cole* and/or *Elgressy*, teaches that the component server determines whether the “determined component and the additional component are consistent with a given one of the

¹⁵ Although this Petition shows that *Pitzel*, alone or in combination with *Cole* and/or *Elgressy*, teaches the subject matter of claim 3, such a showing is not required since claim 3 is a method claim whose single step is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

known acceptable configurations.” See Sections VII.B.2.d-e, and VII.B.3, *supra*. Knutson ¶ 187 (EX1006).

Pitzel discloses that, when a version of a requested component is compatible with the client conditions, the component server determines whether the version of the requested component can be located. See *Pitzel*, 10:13-25 (EX1005); see also Section VII.B.2, *supra*. If it can be located, the component server “creates an upgrade response message to the client computer” that “identif[ies] the location of the computer[] that [is] hosting the requested component[.]” *Pitzel*, 10:21-25 (EX1005); see also Section VII.B.2, *supra*. The client computer then “performs an iterative process for downloading” the component, which can be downloaded from the component server. *Pitzel*, 3:37-39, 10:25-30, 10:55-60, 11:13-23 (EX1005). Thus, *Pitzel* discloses “downloading the determined component to the electronic device if the determined component and the additional component are consistent with a given one of the known acceptable configurations.” Knutson ¶¶ 186, 188 (EX1006).

5. Claims 4 and 14 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “compar[[ing]/[e]] the determined[,] component and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device”

As discussed above in Section VII.B.2.d, *Pitzel*, in combination with *Cole* and/or *Elgressy*, teaches comparing the determined component “and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device.” Knutson ¶ 191 (EX1006).

b) “generat[[ing]/[e]] information indicative of a denial of the reconfiguration request if the determined component and the additional component are consistent with a given one of the known unacceptable configurations”¹⁶

As discussed above in Section VII.B.2.d-e, *Pitzel* discloses that the component server determines whether locations of versions of requested components compatible with the client computer can be identified. *See Pitzel*, 10:13-17 (EX1005). If they cannot, the upgrade request is denied. *See Pitzel*, 10:17-20 (EX1005). Thus, if compatible versions of the requested components

¹⁶ Although this Petition shows that *Pitzel*, alone or in combination with *Cole* and/or *Elgressy*, teaches this limitation of claim 4, such a showing is not required since claim 4 is a method claim and this limitation recites a single step that is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

cannot be located, the component server denies the upgrade request (“generates information indicative of a denial of the reconfiguration request if the determined component and the additional component are consistent with a given one of the known unacceptable configurations”). Knutson ¶ 192 (EX1006).

Moreover, as discussed above in Section VII.B.2.d, *Elgressy* discloses preventing an EO from passing to a client when the computer resources the EO requires are included in a list of resources prohibited for use on that client. Knutson ¶ 193 (EX1006). As further discussed above in Section VII.B.2.d, it would have been obvious to a POSA to modify the combination of *Pitzel* and *Cole* to utilize a table that identifies known inappropriate configurations, such as taught by *Elgressy*. *Id.*

6. Claims 5 and 15 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “*compar[[ing]/[e]] the determined component and information specifying at least one additional component currently implemented in the electronic device with [[the]/[a]] list of known unacceptable configurations for the electronic device*”

As discussed above, *Pitzel*, alone or in combination with *Cole* and/or *Elgressy*, teaches comparing the determined component “and information specifying at least one additional component currently implemented in the electronic device with the list of known unacceptable configurations for the electronic device.” See Sections VII.B.2.d, VII.B.5.a, *supra*. Knutson ¶ 196

(EX1006).

b) “generat[[ing]/[e]] information indicating that the requested reconfiguration is unknown if the determined component and the additional component are not consistent with a given one of the known acceptable or unacceptable configurations”¹⁷

As discussed above in Section VII.B.2.d, *Pitzel*, in combination with *Cole* and/or *Elgressy*, teaches storing and utilizing a list of known compatible configurations and storing and utilizing a list of known incompatible or otherwise inappropriate configurations. Knutson ¶ 197 (EX1006). Any client condition that does not match a component’s metadata on either of these lists would be a client condition for which the appropriateness of the component is unknown. *Id.* It would have been obvious to a POSA at the time of the alleged invention to generate information indicating that a requested reconfiguration is unknown when the client’s conditions do not match the metadata of a component in lists of known

¹⁷ Although this Petition shows that *Pitzel*, in combination with *Cole* and/or *Elgressy*, teaches this limitation of claim 5, such a showing is not required since claim 5 is a method claim and this limitation recites a single step that is conditional. *See Ex parte Schulhauser*, Appeal 2013-007847 (PTAB Apr. 28, 2016) (precedential) (prior art need not disclose conditional steps of method claim to establish obviousness).

appropriate and inappropriate configurations. *Id.* Indeed, identifying an unknown condition in computer software is a practice computer programmers used for many years prior to the filing date of the '088 patent. *Id.* A POSA would have been motivated to make such a modification to make software more robust and less error-prone, and to address situations not matching the known situations the software is programmed to handle. *Id.*

7. Claims 6 and 16 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “transmit[ting] in response to the reconfiguration request a list of additional components required in the electronic device in order to implement the reconfiguration [request]”

Pitzel discloses that the component server can “determine one or more additional components that are necessary for proper operation in addition to those components [] request by the client computer.” *Pitzel*, 9:66-10:2 (EX1005). For example, “one of the requested components may be dependent on the existence of another component in the client computer [] for proper operation.” *Id.*, 10:2-5 (EX1005). The component server can “examine the client conditions to determine the existence of necessary components, and if missing, supply these components in addition to the requested components.” *Id.*, 10:5-12 (EX1005). These components on which the requested components depend are “additional components required in the electronic device in order to implement the reconfiguration.” *Knutson* ¶ 201 (EX1006). *Pitzel* discloses that an upgrade response message is sent to the client

that identifies the location of computers that host the components. *Pitzel*, 10:21-25 (EX1005). This upgrade response message is a “list” of components, which includes the additional components. Knutson ¶¶ 200, 201 (EX1006).

8. Claims 7 and 17 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “wherein the information specifying at least one additional component currently implemented in the electronic device includes identifiers of each of the components in a set of components currently implemented in the electronic device”

Pitzel discloses that client conditions can include information related to the software or hardware components of the client, such as the name or version number of the operating system, the existence of one or more other components, product information, distribution codes, and a unique identifier associated with a microprocessor. *See, Pitzel*, 4:41-52, 6:40-7:11, 8:64-9:22, 1:43-47, 2:3-5 (EX1005). Thus, *Pitzel* discloses that the client conditions includes identifiers of “each of the components in a set of components currently implemented in the electronic device.” Knutson ¶¶ 204, 205 (EX1006).

9. Claims 8 and 18 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “wherein the identifiers of each of the components in the set of components are included in the reconfiguration request [transmitted by the electronic device]”

As discussed above in Section VII.B.8, *Pitzel* discloses that the client conditions includes identifiers of “each of the components in a set of components

currently implemented in the electronic device.” *Pitzel* further discloses that the client conditions are sent from the client to the component server as part of the upgrade request (the “reconfiguration request”). *See Pitzel*, 7:36-43, 7:49-56, 9:23-28 (EX1005). Knutson ¶¶ 208, 209 (EX1006).

10. Claims 9 and 19 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “wherein the reconfiguration request comprises a request for an upgrade of at least one of a software component and a hardware component of the electronic device”

Pitzel discloses that the “present invention advantageously allows users to upgrade software.” *Pitzel*, 11:23-34 (EX1005). The “system automatically downloads a software component and automatically selects the appropriate version of software that is compatible with the user’s computer.” *Id.*, 11:34-37 (EX1005). Knutson ¶¶ 212, 213 (EX1006).

A POSA would recognize that the installation of new software in the memory of a computer necessarily involves a modification (“upgrade”) of that memory. Knutson ¶ 214 (EX1006). Thus, the upgrade request is also a “request for upgrade of ... a hardware component of the electronic device.” Knutson ¶¶ 212, 214 (EX1006).

11. Claims 10 and 20 are obvious in view of *Pitzel*, *Cole*, and *Elgressy*

a) “wherein the reconfiguration request is received from the electronic device over a network connection established [[with a reconfiguration manager implementing the receiving, determining, comparing and generating steps]]/

with a reconfiguration manager which includes the memory and processor]]”

As discussed above, *Pitzel* discloses a component server (“reconfiguration manager”) that receives an “upgrade request” from the client (“electronic device”). *See* Section VII.B.2.b, *supra*. As further discussed above, *Pitzel*, alone or in combination with *Cole* and *Elgressy*, teaches that the component server includes a memory and a processor and implements the “receiving,” “determining,” “comparing,” and “generating” steps. *See* Section VII.B.2, *supra*. Knutson ¶¶ 217, 218 (EX1006).

IPR2019-00056
Patent 6,467,088

VIII. CONCLUSION

For the foregoing reasons, Petitioner requests *inter partes* review of claims 1-21 of U.S. Patent No. 6,467,088.

Dated: October 17, 2018

Respectfully Submitted,

/Larissa S. Bifano/
Larissa S. Bifano
Registration No. 59,051

James M. Heintz
Registration No. 41,828

Michael Van Handel
Registration No. 68,292

IPR2019-00056
U.S. Patent No. 6,467,088

Table of Exhibits for U.S. Patent 6,467,088 Petition for *Inter Partes* Review

Exhibit	Description
1001	U.S. Patent No. 6,467,088
1002	U.S. Patent No. 5,752,042 (“ <i>Cole</i> ”) (filed on June 7, 1996, published on May 12, 1998)
1003	PCT Application Publication No. WO 97/30549 (“ <i>MacInnis</i> ”) (published on August 21, 1997)
1004	U.S. Patent No. 6,449,723 (“ <i>Elgressy</i> ”) (filed on October 30, 1998, published on September 10, 2002)
1005	U.S. Patent No. 7,062,765 (“ <i>Pitzel</i> ”) (filed on May 25, 1999, published on June 13, 2006)
1006	Declaration of Charles D. Knutson, Ph.D
1007	File History, ’607 application, Office Action (02/14/2002)
1008	File History, ’607 application, Response (05/07/2002)
1009	File History, ’607 application, Notice of Allowance (07/29/2002)

IPR2019-00056
U.S. Patent No. 6,467,088

CERTIFICATION UNDER 37 CFR § 42.24(d)

Under the provisions of 37 CFR §42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,812 words, which is less than the 14,000 allowed under 37 CFR §42.24(a)(i).

Dated: October 17, 2018

Respectfully Submitted,

/Michael Van Handel/
Michael Van Handel
Reg. No. 68,292

DLA Piper LLP (US)
33 Arch Street, 26th Floor
Boston, Massachusetts 02110
Tel: (617) 406-6000

IPR2019-00056
U.S. Patent No. 6,467,088

CERTIFICATE OF SERVICE

I hereby certify that on October 17, 2018, I caused a true and correct copy of the foregoing materials:

- Petition for *Inter Partes* Review of U.S. Patent No. 6,467,088 under 35 U.S.C. § 312 and 37 C.F.R. § 42.104
- Exhibit List
- Exhibits for Petition for *Inter Partes* Review of U.S. Patent No. 6,467,088 (EX1001 – EX1009)
- Power of Attorney
- Word Count Certification under 37 C.F.R. § 42.24(d)

to be served via Express Mail on the following correspondent of record as listed on PAIR:

Philips Intellectual Property & Standards
465 Columbus Avenue
Suite 340
Valhalla, NY 10595

/Michael Van Handel/
Michael Van Handel
Reg. No. 68,292